02-23-00

A

02/22/00

jc759 U.S. PTO

mom

Please type a plus sign (+) inside this box → +

PTO/SB/05 (1/98)
Approved for use through 09/30/00. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.	X-560 US
First Named Inventor or Appl. Identifier	Carol A. Fields
Title	System and Method for Assisting in the Development and Integration of Reusable Circuit Designs
Express Mail Label No.	EH864833542US

jc511 U.S. PTO 09/510203 02/22/00

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

1. [X] Fee Transmittal Form (e.g., PTO/SB/17) (Submit an original, and a duplicate for fee processing)
2. [X] Specification [Total Pages 20] (preferred arrangement set forth below)
   - Descriptive title of the Invention
   - Cross References to Related Applications
   - Statement Regarding Fed sponsored R & D
   - Reference to Microfiche Appendix
   - Background of the Invention
   - Brief Summary of the Invention
   - Brief Description of the Drawings (if filed)
   - Detailed Description
   - Claim(s)
3. [X] Drawing(s) (35 USC 113) [Total Sheets 3]
4. Oath or Declaration [Total Pages 2]
   a. [X] Newly executed (original or copy)
   b. [ ] Copy from a prior application (37 CFR 1.63(d)) (for continuation/divisional with Box 17 completed) [Note Box 5 below]
      i. [ ] DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
5. [ ] Incorporation By Reference (useable if Box 4b is checked) The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.
6. [ ] Microfiche Computer Program (Appendix)
7. [ ] Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary)
   a. [ ] Computer Readable Copy
   b. [ ] Paper Copy (identical to computer copy)
   c. [ ] Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

8. [X] Assignment Papers (cover sheet & document(s))
9. [ ] 37 CFR 3.73(b) Statement (when there is an assignee) [ ] Power of Attorney
10. [ ] English Translation Document (if applicable)
11. [ ] Information Disclosure Statement (IDS)/PTO-1449 [ ] Copies of IDS Citations
12. [ ] Preliminary Amendment
13. [X] Return Receipt Postcard (MPEP 503) (Should be specifically itemized)
14. [ ] *Small Entity Statement(s) (PTO/SB/09-12) [ ] Statement filed in prior application, Status still proper and desired
15. [ ] Certified Copy of Priority Document(s) (if foreign priority is claimed)
16. [ ] Other: ....

* A new statement is required to be entitled to pay small entity fees, except where one has been filed in a prior application and is being relied upon

17. If a CONTINUING APPLICATION, check appropriate box and supply the requisite information below and in a preliminary amendment:
[ ] Continuation [ ] Divisional [ ] Continuation-in-part (CIP) of prior application No: ________
Prior application information: Examiner ________ Group / Art Unit: ________

18. CORRESPONDENCE ADDRESS

[ ] Customer Number or Bar Code Label (Insert Customer No. or Attach bar code label here) or [X] Correspondence address below

Name	Edel M. Young				
	XILINX, INC.				
Address	2100 Logic Drive				
City	San Jose	State	California	Zip Code	95124
Country	US	Telephone	408-879-4969	Fax	408-377-6137

Name (print/Type)	Edel M. Young	Registration No. (Attorney/Agent)	32,451
Signature	Edel M Young	Date	February 22, 2000

SYSTEM AND METHOD FOR ASSISTING IN THE DEVELOPMENT AND INTEGRATION OF REUSABLE CIRCUIT DESIGNS

Carol A. Fields
Anthony D. Williams

FIELD OF THE INVENTION

The present invention generally relates to electronic design tools, and more particularly to assisting in the creation and usage of design modules that are amenable for reuse in other designs.

BACKGROUND

Increasingly complex functions are being implemented in ASIC and field programmable gate arrays (FPGAs) given recent advances in silicon technology. Design trends are shifting toward system-level integration in order to reduce the time-to-market and reduce development costs.

System-level integration relies on reuse of previously created designs, either from within an enterprise or from a commercial provider. The engineering community sometimes refers to these previously created designs as “ design modules” , “ cores” or “ IP” (intellectual property). As on-chip processors and large functional blocks become increasingly common, vendors are making complex design modules available for general usage, and companies are making modules for reuse within the respective organizations. These design modules are then integrated into larger systems by end-users.

For reusable design modules to be successful, they must be easy to use. Design modules that are amenable to reuse must be well documented, possess sufficient parameterization, have an understandable structure, follow certain coding guidelines, and be extensible for future usage. Reusable design modules should also have a well-planned and documented testbench. To varying degrees, many commercial and internal design modules satisfy these

EXPRESS MAIL NO: EH864833542US

objectives. However, the process and tools used to create a design module will often limit the ease and extent to which the objectives can be satisfied.

A method that address the aforementioned problems, as well as other related problems, is therefore desirable.

SUMMARY OF THE INVENTION

In various embodiments, the invention provides a method and system for developing a reusable electronic circuit design module and using the design module in a debug mode. In one embodiment, the functional design elements comprising a design module are entered into a database along with documentation elements that describe the design elements. The functional design elements are linked with selected ones of the documentation elements in the database. A testbench is simulated with the design module, and the generated results are stored in a database and linked with the functional design elements. By linking the design elements, documentation, translation results, and simulation results, the characteristics of the design module are easily ascertained by a designer who is reusing the design module.

In another embodiment, a system includes a database, a design inspector, a debugging-support module, and a functional simulator. The database is arranged for storage of the design elements and documentation elements, and the design inspector is coupled to the database. The design inspector links the functional design elements with selected ones of the documentation elements. The debugging-support module is coupled to the simulator and to the database, and generates a netlist from the design module, wherein the netlist is suitable for simulation. The functional simulator is coupled to the debugging-support module and simulates a testbench with the design module, whereby simulation results are generated. The

simulation results are entered in the database by the debugging-support module and thereafter linked with the design elements.

The above summary of the present invention is not intended to describe each disclosed embodiment of the present invention. The figures and detailed description that follow provide additional example embodiments and aspects of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Various aspects and advantages of the invention will become apparent upon review of the following detailed description and upon reference to the drawings in which:

FIG. 1 is a block diagram of a system for creating reusable design modules in accordance with one embodiment of the invention;

FIG. 2 is a flowchart of an process for developing reusable logic in accordance with one embodiment of the invention; and

FIG. 3 is a flowchart of a process for integrating reusable logic into a design in accordance with another embodiment of the invention.

DETAILED DESCRIPTION

The present invention is believed to be applicable to the process of creating reusable design modules for a variety of electronic circuit technologies. An example environment for creating design modules for field programmable gate arrays (FPGAs) is used to describe the various embodiments of the invention. While the present invention is not so limited, an appreciation of the present invention is presented by way of specific examples involving FPGAs.

In accordance with various embodiments of the invention, a design inspector tool works in conjunction with a design entry tool to assist in creating design modules

EXPRESS MAIL NO: EH864833542US

that are easy to reuse. Specifically, the design inspector processes a design module to determine whether there is documentation that conforms to specified criteria and whether the design module conforms to other specified design rules. Where the design module fails to conform to the specified criteria, a report is provided so that the deficiency can be remedied. In addition, the design elements that comprise the design module, the associated documentation, translation results, and simulation results are linked in a database to assist in understanding particular aspects of the design in view of simulation results.

The present invention supports two modes of operation. The first mode is a design mode where the first instance of a design module is created for reuse. The initial design module may or may not be part of a fully operational system or integrated with other modules. The second mode is an integration mode where a reusable design module is integrated with other design modules to create a netlist. The netlist is translated and then simulated, wherein the translation results and the simulation results are correlated with the design modules and associated documentation. After creating a physical implementation from the netlist, the physical implementation is simulated, and the simulation results are correlated with the design modules and associated documentation. The final design, as well as the design module as modified, can be saved in a central depository for further reuse.

By inspecting for desired documentation and desired design characteristics at appropriate stages and creating a database of design modules, documentation, netlist, and simulation results, easy-to-reuse design modules can be created.

FIG. 1 is a block diagram of a system for creating reusable design modules in accordance with one embodiment of the invention. System 100 includes database 102, which

includes various design elements and associated documentation elements. Design inspector 104, when activated by the user, examines the design elements for various predefined characteristics. For example, inspector 104 inspects for proper documentation, along with desirable parameterization, security, revision control, hierarchical arrangement, partitioning, and adherence to predetermined design rules, all as embodied in logic within the design inspector.

Design entry tool 106 is used to initially create a design module, and in different embodiments may be an RTL editor, state machine editor, or schematic capture tool, for example. The user interacts with the various components via user interface logic 108, and design inspector 104 can be invoked either via design entry tool 106 or by the user via user interface 108.

Database 102 is created by design entry tool 106 via design inspector 104. Database 102 includes several design elements 110 that comprise a design module. Associated with the design elements are various documentation elements 112. The documentation elements may include, among other items, a datasheet, a functional block diagram, a state diagram, descriptions of blocks, intended usage, parameters that can be modified, effects of modifying parameters, expansion effects, descriptions of input and output signals, description of processing as well as other design descriptive information.

Database 102 not only provides a mechanism to reference documentation associated with various elements of a design, but also provides a mechanism to correlate the design elements and associated documentation with a netlist 114 and physical implementation 116, along with the functional simulation results 118 and physical simulation results 120. Since a design module will undergo various translations, e.g., synthesis, in progressing toward simulation, the translation results are correlated with design elements and

EXPRESS MAIL NO: EH864833542US

associated documentation in order to facilitate debugging a design. Thus, based on the simulation results, a designer can easily reference design information and documentation associated with the implementations.

When a designer is ready to begin testing and debugging a design, a netlist 114 is created. The netlist, along with a suitable testbench (not shown) is used to test the functionality of the design. The testbench is stored in a file which is correlated with the design elements. The design is simulated for functional correctness using functional simulator 122 and applying the testbench. Error and warning messages are written to the database and correlated with the design by debug support logic 124. Example functional simulators include Modelsim from Model Technology and VSS from Synopsys. The data resulting from the simulation is written to functional simulation results file 118.

In order to facilitate debugging, debugging support logic 124 correlates the simulation results from functional simulator 122 with design elements 110 and documentation elements 112 from database 102. For example, if design entry tool 106 is a state machine editor, errors are correlated to possible state or state transitions containing the error. For a schematic capture design entry tool, a correlation of the error to the vicinity of the schematic containing the error is provided. Correlation of simulation results to design elements and documentation elements enables display of the documentation via user interface 108.

Debugging support logic 124 tracks and correlates how the design module is translated. For example, HDL design constructs are traced from high-level design to design elements. In schematic C, any changes made by a translation tool are tracked.

After the errors discovered in the functional simulation have been corrected, the design can be physically implemented via implementation translators 128. Example

translators include DC2NCF, NGD2VHDL, and NGDZVER translators from Xilinx. Physical implementation 116 is a netlist, for example.

Physical simulator 126 runs a simulation using a predefined testbench and interfaces with debugging support logic 124 to log the physical simulation results 120. The physical simulation results are also correlated with the design elements and documentation of database 102. By tracking the translation of the design elements from high level design through translation to the physical implementation, the constructs of the high-level design are correlated with elements in the physical implementation. This correlation is then used by debugging support logic 124 to correlate the physical simulation results to the design and documentation elements. In FPGA technology, the representation of the design may differ from the implementation. Tracking the changes as the design elements progress through the translations and correlating the changes to the documentation assists in reuse of a module.

FIG. 2 is a flowchart of a process for developing reusable logic in accordance with one embodiment of the invention. The process generally entails entering and then inspecting a design module, modifying the design to correct any deficiencies and then re-inspecting. The design module is also inspected for a proper level of documentation. The design elements that comprise the design module and documentation elements that are associated with the design elements are stored in a database for future reference. In addition to the design module, a testbench is created for use with the simulator. The testbench is also associated with the design module for future use.

At step 202, a design module is created using a design entry tool that is adapted to provide the functions of the present invention. A design script is also created by the designer. The design script contains the directives that specify which tools to run, the order in which the tools are

to run, as well as options and environment variables for the tools. The design script is stored in a separate file.

The design module and script file are inspected at step 204 for selected design characteristics. For example, the characteristics may include desired parameterization, adherence to specified design rules, and a suitable hierarchical arrangement of the design. In addition, the design inspector checks that all ranges are enumerated and that there is a consistent number of multi-bit objects.

The design inspector is configured to enforce certain design rules. For example, the rules may include a certain number of spaces for indentation of the code, one statement per line, a maximum of 72 characters/line, use of tabs for tabular layout, no usage of abbreviations, capitalization rules, usage of suffixes, reserved uppercase for constants, usage of underscore character for separation of compound words, no misuse of reserved words, usage of " _n" for active low symbols, usage of " clk" prefix for clock signals, usage of a common clock name for all derived and synchronized clock signals, usage of symbolic names to define states in a finite state machine, proper usage of filename extensions for identification of file type, and language specific design rules (e.g., process labels).

For a desired hierarchical arrangement, the design inspector checks whether there are constants that can be changed to variables defined at the top sub-module level, and that there are no more than three levels of nesting. In addition, the design inspector checks that names are preserved across interfaces and hierarchical boundaries. The design inspector may also be programmed to provide a graphical representation of the present hierarchical arrangement to assist in tracking, adding, and deleting sub-modules.

Where the design module fails to conform to the selected characteristics, a report is provided to the designer at step 206. In response to the report by the

EXPRESS MAIL NO: EH864833542US

design inspector, the user may modify the design and then re-inspect (step 208). The modify and re-inspect cycle may be repeated as many times as required to achieve desired design goals.

As part of the re-inspection, the design inspector tracks which of the design elements are changed from the prior inspection and reports which additional design elements are dependent on such changes. For example, changing the value of a constant causes the design inspector to report all sub-modules which use the constant.

At step 210, the design inspector is invoked to check that documentation has been entered for the design module and that the documentation conforms to characteristics imposed by the design inspector. At step 212, the documentation can be entered and/or updated in response to the report provided by the design inspector. Thereafter the design modules can be re-inspected.

There are numerous types of documentation that may be required. Examples include: copyright and confidentiality notices, brief descriptions, revision numbers, past and current authors, change histories, functionality descriptions, descriptions of code structure, variable definitions and side effects, enumeration of valid input data ranges, identifications of parameters not intended to be easily modified, and cross references to applicable industry standards. In addition it may be desirable to require documentation as to the implementation implications of modifying various parameters. For example, expanding a 16x16 multiplier may cause the multiplier to wrap into several columns of configurable logic blocks (CLBs) in a field programmable gate array (FPGA).

Once the desired documentation has been entered, the documentation elements and design elements that comprise the design module are linked in a database. The database provides easy future reference to the design and

documentation, such as when the design has been implemented and has undergone several translation and simulation steps.

At step 216, a testbench is entered by the user using conventional tools. The testbench will also be referred to in this document as " simulation elements" . Simulation elements are similar to design elements except that they are generated for the purpose of testing the functionality of the design. The testbench is inspected at step 218 by the design inspector, and at step 220, the characteristics of the testbench are reported to the user. The testbench is modified, as may be necessary, at step 222 and then re-inspected. The modify/re-inspect step may be repeated as necessary to correct any deficiencies.

At step 224, the documentation for the testbench is created, and the testbench is re-inspected. Example documentation for testbenches includes: documenting the features included in the testbench, enumeration of assumptions and omissions, description of an input sequence, description of expected output behavior, and a description of anticipated design flow. The elements that comprise the testbench and the associated documentation are added to the design database at step 226. Once the design and testbench have been suitably structured and documented and added to the database, the process is complete. The design module is then in a form that is amenable to reuse.

FIG. 3 is a flowchart of a process for integrating reusable logic into a design in accordance with another embodiment of the invention. The process generally entails integrating a design module, which was created in accordance with the process of FIG. 2, with other design modules to create a netlist. The resulting logic can then be documented, structured, and inspected to create a design module that can be saved for future integration with still other design modules.

At step 302, the desired design module is retrieved from a central depository. The central depository may be a

tree structure of files containing design modules, associated documentation, and test benches. The documentation elements and testbench associated with the selected design module are retrieved at step 304.

The selected design module is modified at step 306 in a manner that is suitable for integration with other design modules. The nature of the modifications to the selected design module is dependent on the function and interfaces of the modules with which the selected module is being integrated. The documentation elements and testbench are also modified as may be necessary.

At step 308, the new design (selected design module as integrated with other design modules) is translated into a netlist. The netlist is then written to a file at step 310. At step 312, the translation results are correlated with the design elements, documentation elements, and testbench elements of the central depository database. In order to correlate elements of the netlist with the original design elements, the elements generated during the translation process are associated in a database with the respective design elements from which they were generated.

The functional design is simulated at step 314, and the simulation results are written to a file at step 316. At step 318, the simulation results are correlated with the design elements, documentation elements, and testbench elements in the central depository database. The correlation provides a mechanism for the designer to trace particular portions of the simulation results back to the original design elements and associated documentation. If an error is discovered during functional simulation, the offending design elements can be changed, saved, re-implemented, and simulated as needed.

At step 320, the functional design is translated into a physical design using conventional tools adapted to work in conjunction with the present invention. The physical implementation is written to a file at step 322, and the

physical translation results are correlated with the design elements and documentation elements at step 324.

The following example illustrates the correlation of design elements to physical translation results. The following snippet of VHDL code is taken from a design that implements a state machine.

```
constant IDLE_CNT   : INTEGER := 8;
...

when STABILIZATION_WAIT =>
ISOLATE          <= TRUE;
TIMER_TICK       <= CLK_1K;
if(IDLE_DONE = TRUE) then
     NEXT_STATE <=LINK_WAIT;
elseif (SCARRIER_PRESENT = TRUE) then
     NEXT_STATE <= VALID_START;
else
     NEXT_STATE <= STABILIZATION_WAIT;
endif;
...
process (COUNT)
begin
     if (COUNT < IDLE_CNT) then
          IDLE_DONE <= '0';
     else
          IDLE_DONE <= '1';
     end if;
end process;
...
```

In documenting this portion of the design, the designer creates documentation for the design element STABILIZATION_WAIT indicating that a constant IDLE_CNT controls the variable IDLE_DONE, which influences what the transition to the LINK_WAIT state. Further documentation that is associated with IDLE_DONE explains the usage of IDLE_DONE and the implications of changing the constant value.

The following snippet of code sets forth the physical translation results that is generated from the VHDL set forth above.

```
defparam \current_state(0)/G .INIT = 16'hECCC;
X_LUT4 \current_state(0)/G (
     .ADR0 (un26_count_3),
     .ADR1 (current_state[6]),
     .ADR2 (current_state[0]),
```

```
            .ADR3 (SCARRIER_PRESENT_c),
            .O (\current_state[0]/GROM )
      );
      defparam \current_state(0)/F .INIT = 16'h0001;
      X_LUT4 \current_state(0)/F (
            .ADR0 (current_state[1]),
            .ADR1 (current_state[6]),
            .ADR2 (current_state[2]),
            .ADR3 (current_state[0]),
            .O (\current_state[0]/FROM )
      );
      X_BUF \current_state(0) /XUSED (
            .I (\current_state[0]/FROM ),
            .O (ISOLATE_iv)
      );
      X_FF \current_state(0)/FFY/ASYNC_FF (
            .I (\current_state[0]/GROM ),
            .CLK (RIC_CLK_c),
            .CE (VCC),
            .SET (GND),
            .RST (\current_state[0]/FFY/ASYNC_FF_GSR_OR ),
            .O (current_state[0])
      );
```

It can be seen in the physical translation that the state STABILIZATION_WAIT has been renamed. Without correlation between the physical translation results and the functional design element, the designer would be left to determine which elements in the physical translation correspond to the elements in the design. In this example, current_state[0] corresponds to STABILIZATION_WAIT. To assist the designer during test and debug activities, STABILIZATION_WAIT is correlated with current_state[0] by the translators and debugging support logic. Thus, when performing physical simulation, results that reference current_state[0] can be traced by the designer to the state STABILIZATION_WAIT, which has linked documentation that references the constant IDLE_CNT.

The correlation of the physical translation results with the original design elements and documentation elements is especially helpful in the context of designs for programmable logic devices (PLDs). Example PLDs include field programmable gate arrays (FPGAs) that are available from Xilinx. The FPGA-based physical implementation of a

design may have little or no resemblance to the original design module. Thus, it is beneficial to correlate elements of the physical translation with the originating design elements and associated documentation elements.

The physical implementation is simulated at step 326, and the simulation results are written to a file at step 328. At step 330, the simulation results are correlated with the design elements and documentation elements. If redesign is necessary, the appropriate design elements can be modified, and the process can be repeated beginning at step 308.

At step, 332, the new design is subjected to the process of FIG. 2. That is, the new design is processed by the design inspector to determine whether the selected design rules and documentation requirements have been adhered to.

Accordingly, the present invention provides, among other aspects, a system and method for creating reusable design modules for electronic circuits. Other aspects and embodiments of the present invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and illustrated embodiments be considered as examples only, with a true scope and spirit of the invention being indicated by the following claims.

CLAIMS

What is claimed is:

1. A computer-implemented method for developing a reusable electronic circuit design module, wherein the design module is comprised of one or more functional design elements comprising the design module, comprising:

entering the functional design elements into a database;

entering documentation elements into the database;

linking the functional design elements with selected ones of the documentation elements;

simulating a testbench with the design module, whereby simulation results are generated;

storing the simulation results in the database; and

linking the simulation results with the functional design elements.

2. The method of claim 1, further comprising:

translating the functional design elements into a netlist; and

linking elements of the netlist with selected ones of the functional design elements.

3. The method of claim 2, further comprising:

translating the functional design elements into a physical implementation; and

linking elements of the physical implementation with selected ones of the functional design elements.

4. The method of claim 1, further comprising:

entering simulation elements in the database; and

linking the simulation elements to associated ones of the design elements.

5. The method of claim 4, further comprising:

EXPRESS MAIL NO: EH864833542US

entering documentation for a design script in the database; and

linking the documentation of the design script to the design elements comprising the design module.

6. The method of claim 4, further comprising:

entering documentation for the simulation elements in the database; and

linking the documentation for the simulation elements with associated ones of the simulation elements.

7. The method of claim 6, further comprising:

inspecting the functional design elements and simulation elements for associated documentation; and

reporting documentation deficiencies in association with the functional design elements and simulation design elements.

8. The method of claim 1, further comprising:

inspecting the functional design elements for associated documentation; and

reporting documentation deficiencies in association with the functional design elements.

9. The method of claim 1, further comprising:

inspecting the functional design elements for undesirable design characteristics; and

reporting the undesirable design characteristics found in the functional design elements.

10. The method of claim 9, further comprising:

inspecting the functional design elements for undesirable hierarchical characteristics; and

reporting discovered ones of the undesirable hierarchical characteristics.

EXPRESS MAIL NO: EH864833542US

11. The method of claim 9, further comprising:

inspecting the functional design elements for adherence to predefined design rules; and

reporting violations of the design rules.

12. The method of claim 11, further comprising providing assistance in specifying the design rules for the functional design elements.

13. The method of claim 9, further comprising:

monitoring changes made to the functional design elements; and

indicating which of the functional design elements are dependent on the changes.

14. The method of claim 1, further comprising:

translating the functional design elements into a physical implementation; and

linking elements of the physical implementation with selected ones of the functional design elements.

15. The method of claim 1, further comprising requiring specification of parameters at a top level of a hierarchy of the design module.

16. The method of claim 1, further comprising displaying the functional design elements linked to errors in the simulation results.

17. The method of claim 16, further comprising displaying documentation elements associated with errors in the simulation results.

18. An apparatus for developing a reusable electronic circuit design module, wherein the design module is

comprised of one or more functional design elements comprising the design module, comprising:

means for entering the functional design elements into a database;

means for entering documentation elements into the database;

means for linking the functional design elements with selected ones of the documentation elements;

means for simulating a testbench with the design module, whereby simulation results are generated;

means for storing the simulation results in the database; and

means for linking the simulation results with the functional design elements.

19. A system for developing a reusable electronic circuit design module, wherein the design module is comprised of one or more functional design elements comprising the design module, comprising:

a database arranged for storage of the design elements and documentation elements;

a design inspector coupled to the database, the design inspector configured and arranged to link the functional design elements with selected ones of the documentation elements;

a debugging-support module coupled to the simulator and to the database, the debugging-support module configured and arranged to generate a netlist from the design module, wherein the netlist is suitable for simulation;

a functional simulator coupled to the debugging-support module, the simulator configured and arranged to simulate a testbench with the design module, whereby simulation results are generated; and

wherein the debugging-support module is further configured and arranged to store the simulation results in

the database and link the simulation results with the functional design elements.

EXPRESS MAIL NO: EH864833542US

SYSTEM AND METHOD FOR ASSISTING IN THE DEVELOPMENT AND INTEGRATION OF REUSABLE CIRCUIT DESIGNS

Carol A. Fields
Anthony D. Williams

ABSTRACT

A system and method for developing a reusable electronic circuit design module are presented in various embodiments. In one embodiment, the functional design elements comprising a design module are entered into a database along with documentation elements that describe the design elements. The functional design elements are linked with selected ones of the documentation elements in the database. A testbench is simulated with the design module, and the generated results are stored in a database and linked with the functional design elements. By linking the simulation results, documentation, and design elements, the characteristics of the design module are easily ascertained by a designer who is reusing the design module.

EXPRESS MAIL NO: EH864833542US

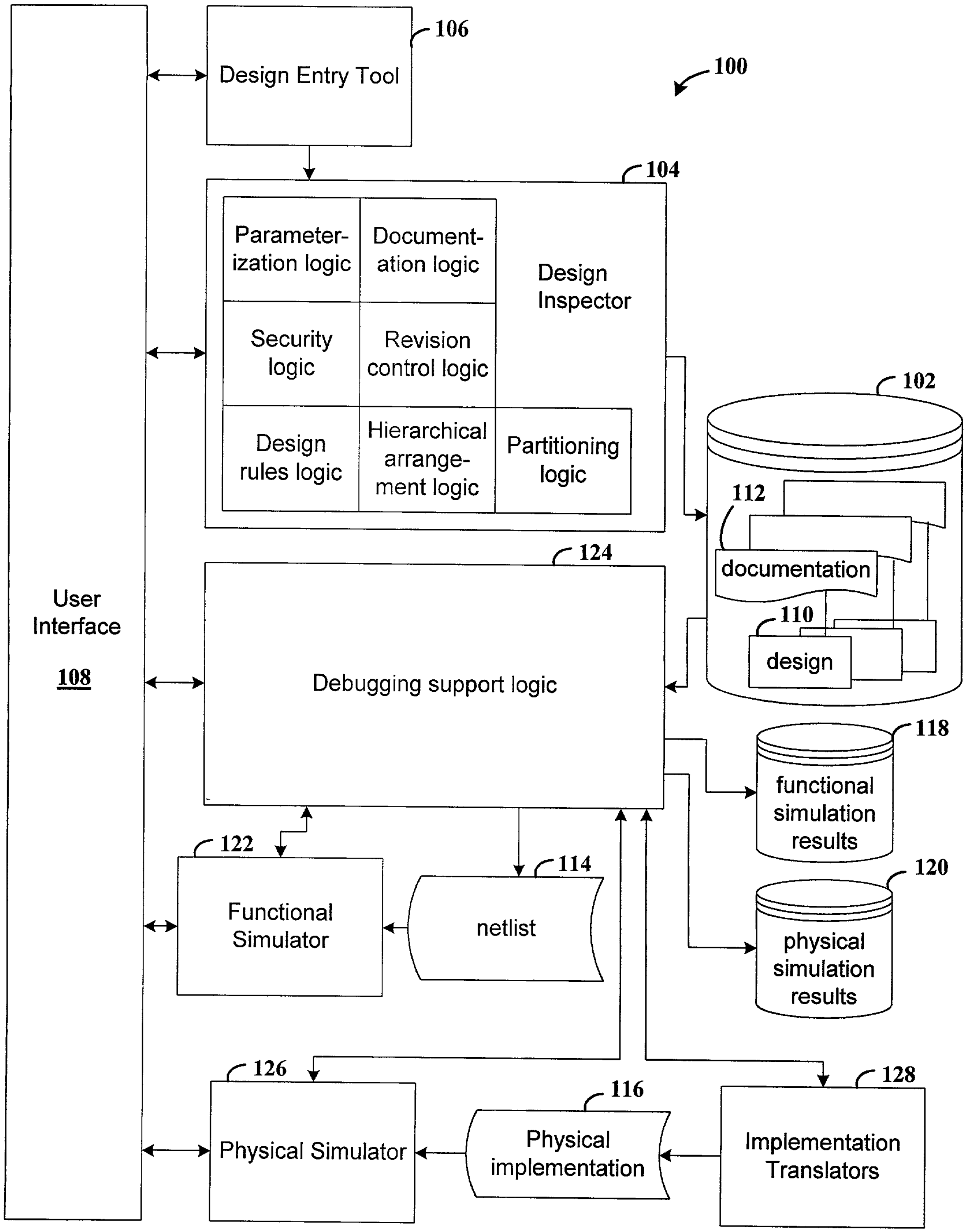

100
Design Entry Tool
106
104
Parameter-ization logic
Document-ation logic
Design Inspector
Security logic
Revision control logic
Design rules logic
Hierarchical arrange-ment logic
Partitioning logic
102
112
documentation
110
design
User Interface
108
124
Debugging support logic
118
functional simulation results
120
physical simulation results
122
Functional Simulator
114
netlist
126
Physical Simulator
116
Physical implementation
128
Implementation Translators

FIG. 1

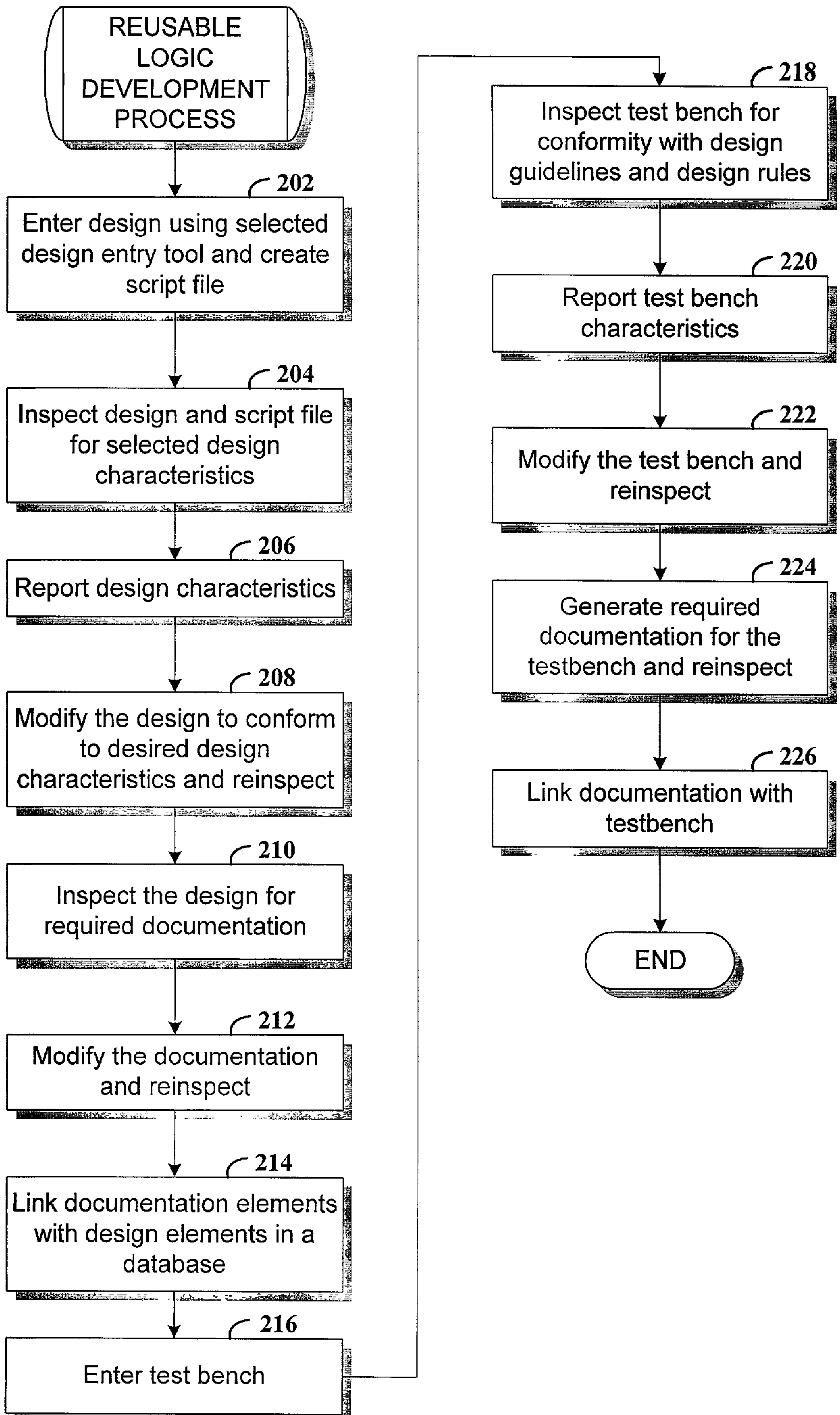
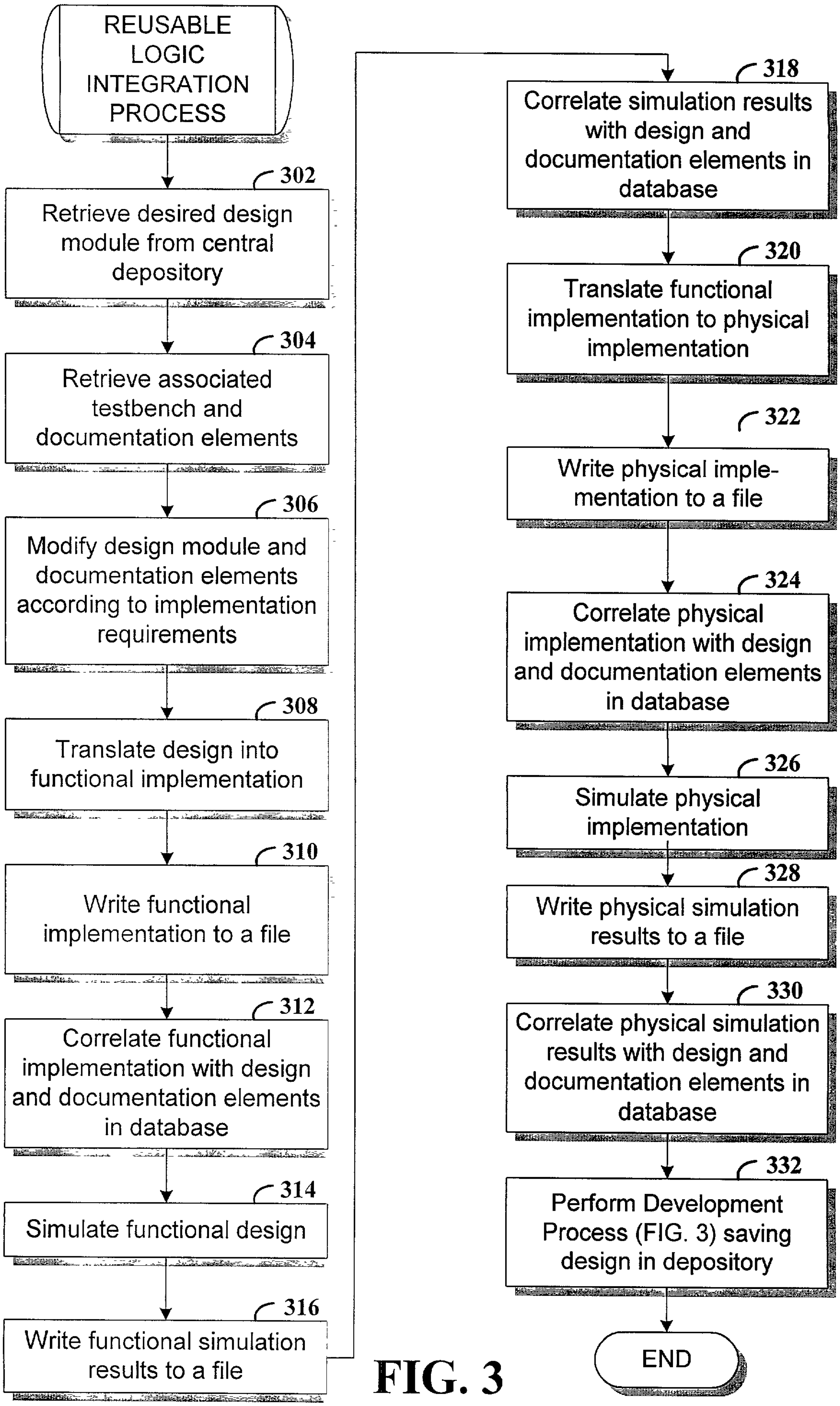

REUSABLE LOGIC DEVELOPMENT PROCESS
202
Enter design using selected design entry tool and create script file
204
Inspect design and script file for selected design characteristics
206
Report design characteristics
208
Modify the design to conform to desired design characteristics and reinspect
210
Inspect the design for required documentation
212
Modify the documentation and reinspect
214
Link documentation elements with design elements in a database
216
Enter test bench
218
Inspect test bench for conformity with design guidelines and design rules
220
Report test bench characteristics
222
Modify the test bench and reinspect
224
Generate required documentation for the testbench and reinspect
226
Link documentation with testbench
END

FIG. 2



REUSABLE LOGIC INTEGRATION PROCESS
302
Retrieve desired design module from central depository
304
Retrieve associated testbench and documentation elements
306
Modify design module and documentation elements according to implementation requirements
308
Translate design into functional implementation
310
Write functional implementation to a file
312
Correlate functional implementation with design and documentation elements in database
314
Simulate functional design
316
Write functional simulation results to a file
318
Correlate simulation results with design and documentation elements in database
320
Translate functional implementation to physical implementation
322
Write physical implementation to a file
324
Correlate physical implementation with design and documentation elements in database
326
Simulate physical implementation
328
Write physical simulation results to a file
330
Correlate physical simulation results with design and documentation elements in database
332
Perform Development Process (FIG. 3) saving design in depository
END

FIG. 3

Docket No. X-560 US

DECLARATION FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below adjacent to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of subject matter (process, machine, manufacture, or composition of matter, or an improvement thereof) that is disclosed and/or claimed and for which a patent is solicited by way of the application entitled

SYSTEM AND METHOD FOR ASSISTING IN THE DEVELOPMENT AND INTEGRATION OF REUSABLE CIRCUIT DESIGNS

which (check)

[X] is attached hereto.
[ ] and is amended by the Preliminary Amendment attached hereto.
[ ] was filed on ________________ as Application Serial No.____________.
[ ] and was amended on ________________ (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified application, including the claims, including portions amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, § 1.56(a).

I hereby claim foreign priority benefits under title 35, United States Code, § 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed	
			Yes	No
(Number)	(Country)	(Day/Month/Year Filed)		
			Yes	No
(Number)	(Country)	(Day/Month/Year Filed)		

I hereby claim the benefit under 35 U.S.C. § 119(e) of any United States provisional application(s) listed below

(Application Number(s))	(Filing Date (MM/DD/YYYY))

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s) listed below and, insofar as any subject matter of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, § 1.56(a) which occurred between the filing date of the prior application(s) and the national or PCT international filing date of this application:

(Application Serial No.)	(Filing Date)	(Status-patented, pending, abandoned)
(Application Serial No.)	(Filing Date)	(Status-patented, pending, abandoned)

Docket No. X-560 US

I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and to transact all business in the United States Patent and Trademark Office connected herewith:

Edel M. Young (32,451), and Lois D. Cartier (40,941) located at Xilinx, Inc., 2100 Logic Drive, San Jose, California 95124, and
LeRoy D. Maunu (35,274) located at Crawford, PLLC, 333 Washington Avenue N, Suite 5000, Minneapolis, Minnesota 55401.

Address all telephone calls to: Edel M. Young at Telephone No. (408) 879-4969

Address all correspondence to: Edel M. Young
Xilinx, Inc.
2100 Logic Drive
San Jose, California 95124

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Title 18, United States Code, § 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of sole or first inventor Carol A. Fields
Inventor's signature Carol A. Fields Date February 22, 2000
Residence 18430 Main Boulevard, Los Gatos, CA 95033 Citizenship United States
Post Office Address 18430 Main Boulevard, Los Gatos, CA 95033

Full name second inventor Anthony D. Williams
Inventor's signature [signature] Date 2/17/00
144 POTOMAC DR, LOS GATOS, CA 95032
Residence ~~3500 Treetops Circle, San Bruno, CA 94066~~ Citizenship United Kingdom
Post Office Address ~~3500 Treetops Circle, San Bruno, CA 94066~~
PO BOX 54149, SAN JOSE, CA 95154-4149

Full name third inventor None
Inventor's signature Date
Residence Citizenship
Post Office Address

Full name fourth inventor None
Inventor's signature Date
Residence Citizenship
Post Office Address

EMY:mom